

```

1  /*
2  Display.h - A simple track GPS to SD card logger. Display module.
3  TinyTrackGPS v0.7
4
5  Copyright © 2019-2021 Francisco Rafael Reyes Carmona.
6  All rights reserved.
7
8  raphael.reyes.carmona@gmail.com
9
10  This file is part of TinyTrackGPS.
11
12  TinyTrackGPS is free software: you can redistribute it and/or modify
13  it under the terms of the GNU General Public License as published by
14  the Free Software Foundation, either version 3 of the License, or
15  (at your option) any later version.
16
17  TinyTrackGPS is distributed in the hope that it will be useful,
18  but WITHOUT ANY WARRANTY; without even the implied warranty of
19  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
20  GNU General Public License for more details.
21
22  You should have received a copy of the GNU General Public License
23  along with TinyTrackGPS. If not, see <https://www.gnu.org/licenses/>.
24  */
25
26  #if ARDUINO >= 100
27    #include "Arduino.h"
28  #else
29    #include "WProgram.h"
30  #endif
31
32  #ifndef Display_h
33  #define Display_h
34
35  #include "config.h"
36
37  #if defined(DISPLAY_TYPE_LCD_16X2)
38    #include <LiquidCrystal.h>
39  #elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
40    #include <LiquidCrystal_I2C.h>
41  #elif defined(DISPLAY_TYPE_SDD1306_128X64)
42    #define U8X8_HAVE_HW_I2C
43    #include <U8x8lib.h>
44    // #include <U8g2lib.h>
45  #endif
46
47  enum Display_Type {
48      SDD1306_128X64,    // Para usar pantalla OLED 0.96" I2C 128x64 pixels
49      LCD_16X2,          // Para usar LCD 16 x 2 carateres.
50      LCD_16X2_I2C       // Para usar LCD 16 x 2 carateres. I2C.
51  };
52
53  class Display {
54  private:
55      byte _offset;
56      byte _width;       // Width pixels or numbers of columns for LCD.
57      byte _height;      // Height pixels os numbers of rows for LCD.
58      Display_Type _screen;
59      #if defined(DISPLAY_TYPE_LCD_16X2)
60      LiquidCrystal* lcd;

```

```

61     #elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
62         LiquidCrystal_I2C* lcd;
63     #elif defined(DISPLAY_TYPE_SDD1306_128X64)
64         //U8G2_SSD1306_128X64_NONAME_1_HW_I2C* u8g2_SSD1306;
65         U8X8_SSD1306_128X64_NONAME_HW_I2C* u8x8_SSD1306;
66     #elif defined(DISPLAY_TYPE_HX1230_96X68)
67         U8G2_HX1230_96X68_1_3W_SW_SPI* u8g2_HX1230;
68     #endif
69
70     public:
71         Display(Display_Type t = SDD1306_128X64);
72         Display() = delete; // Constructor por defecto.
73         Display(const Display&) = delete; // Constructor de copia.
74
75         void start();
76         void clr();
77         void print(int, int, const char[]);
78         void print(int, const char[]);
79         void print(const char[]);
80         void print(const char[], const char[]);
81         void print(const char[], const char[], const char[]);
82         void print(const char[], const char[], const char[], const char[]);
83         void wait_anin(unsigned int);
84         void draw_wait(byte);
85         void print_PChar(byte);
86         void splash(int time_delay = 750);
87         Display_Type display_type(){return _screen;};
88 };
89
90 #endif

```